

UNIVERZA V LJUBLJANI  
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Dejan Dežman

**Upravljanje pospeševalnika delcev**  
**European Spallation Source**

DIPLOMSKO DELO  
NA UNIVERZITETNEM ŠTUDIJU

MENTOR: dr. Andrej Brodnik

Ljubljana 2013



Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

*Besedilo je oblikovano z urejevalnikom besedil  $\text{\LaTeX}$ .*





Št. naloge: 01953 / 2013  
Datum: 2.9.2013

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **DEJAN DEŽMAN**

Naslov: **UPRAVLJANJE POSPEŠEVALNIKA DELCEV EUROPEAN SPALLATION  
SOURCE  
MANAGEMENT OF EUROPEAN SPALLATION SOURCE**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Kot vsaka naprava se tudi pospeševalnik delcev European Spallation Source sestoji iz nadzornega dela in izvajalnega dela. V diplomskem delu preučite kako učinkovito upravljati vhodno izhodne naprave, ki se uporabljajo za izvajanje meritev, ter implementirajte sistem. Primerjajte upravljanje pospeševalnika s klasičnim upravljanjem v računalniških omrežjih ter preverite možnost standardizacije upravljanja pospeševalnika z uporabo obstoječih protokolov.

Mentor:

doc. dr. Andrej Brodnik



Dekan:

prof. dr. Nikolaj Zimic



## IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Dejan Dežman, z vpisno številko **63070069**, sem avtor diplomskega dela z naslovom:

*Upravljanje pospeševalnika delcev European Spallation Source*

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom dr. Andreja Brodnika,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki „Dela FRI“.

V Ljubljani, dne 29. oktobra 2013

Podpis avtorja:





*Zahvaljujem se mentorju dr. Andreju Brodniku za mentorstvo in pomoč pri izdelavi diplomske naloge ter Mihu Votoroviču iz podjetja Cosylab, d. d., za mentorstvo na projektu European Spallation Source.*

*Zahvaljujem se tudi Katarini za vzpodbudo pri zadnjih izpitih ter mami in očetu za potrpežljivost in podporo v času študija.*



# Kazalo

Kratice in simboli

Povzetek

Abstract

<b>1</b>	<b>Uvod</b>	<b>1</b>
<b>2</b>	<b>Upravljanje pospeševalnika European Spallation Source</b>	<b>3</b>
2.1	Pospeševalnik European Spallation Source . . . . .	3
2.2	Integrirani kontrolni sistem . . . . .	5
2.3	Experimental Physics and Industrial Control System (EPICS)	7
2.4	Upravljanje kontrolnega sistema . . . . .	9
<b>3</b>	<b>Osnove upravljanja omrežij</b>	<b>11</b>
3.1	Splošno . . . . .	11
3.2	Standardi . . . . .	12
3.3	Model upravljanja omrežij . . . . .	13
3.4	SNMP . . . . .	15
3.5	SNMPv3 . . . . .	26
<b>4</b>	<b>Izvedba</b>	<b>27</b>
4.1	Razvoj nove naprave . . . . .	27
4.2	Odjemalec konfiguratorja kontrolnega sistema . . . . .	30

## KAZALO

<b>5</b>	<b>Rezultati in razprava</b>	<b>35</b>
5.1	Organizacijski model . . . . .	35
5.2	Informacijski model . . . . .	37
5.3	Komunikacijski model . . . . .	37
<b>6</b>	<b>Zaključek</b>	<b>39</b>

# Kratice in simboli

ASN.1 – Abstract Syntax Notation One

BER – Basic Encoding Rules

CA – Channel Access

CLI – Command-line Interface

CMIP – Common Management Information Protocol

CMIS – Common Management Information Service

DMTF – Distributed Management Task Force

EPICS – Experimental Physics and Industrial Control System

GUI – Graphical User Interface

ICS – Integrated Control System

IEEE – Institute of Electrical and Electronics Engineers

IETF – Internet Engineering Task Force

IOC – Input Output Controller

ISO – International Standard Organization

ITU – International Telecommunication Union

LAN – Local Area Network

MAN – Metropolitan Area Network

MIB – Management Information Base

MIT – Management Information Tree

NSM – Network Management System

OSI – Open Systems Interconnection

PDU – Protocol Data Unit

REST – Representational State Transfer

## *KAZALO*

RMON – Remote Monitoring

SMI – Structure of Management Information

SNMP – Simple Network Management Protokol

TLV – Type Length Value

TMN – Telecommunications Management Network

UDP – User Datagram Protocol

# Povzetek

Diplomska naloga predstavlja implementacijo konfiguratorja kontrolnega sistema za pospeševalnik delcev *European Spallation Source* ter diskusijo o tem, kako bi uporaba standardov na področju upravljanja omrežij pripomogla k izboljšanju implementirane rešitve. Diplomska naloga v prvem delu predstavi pospeševalnik delcev *European Spallation Source* ter na kratko opiše glavne dele kontrolnega sistema. V drugem delu obravnava osnove teorije upravljanja omrežij ter protokol SNMP. V tretjem delu opisuje implementacijo konfiguratorja kontrolnega sistema, v zadnjem delu pa razpravlja o uporabi protokola SNMP v sami implementaciji in o tem, kaj bi njegova uporaba doprinesla k sami rešitvi.

Ključne besede: kontrolni sistem, European Spallation Source, konfigurator kontrolnega sistema, protokol SNMP, navidezna baza MIB





# Abstract

The thesis presents the implementation of control system configurator for particle accelerator European Spallation Source. It also presents the discussion about how would the use of standards in the field of network management contribute to the implementation. The first part describes the particle accelerator European Spallation Source and the main components of the control system architecture. The second part discusses network management basics and describes the SNMP protocol. The third part deals with the implementation of control system configurator and the last part discusses the features of use of SNMP protocol and how it would influence the implemented solution.

Keywords: control system, European Spallation Source, control system configurator, SNMP protocol, MIB



# Poglavje 1

## Uvod

*European Spallation Source* je pospeševalnik delcev, ki bo z metodo sipanja nevtronov omogočal raziskovanje različnih znanstvenih področij. Gradnja se bo začela leta 2013, polno delujoč pa bo leta 2025. Sam pospeševalnik delcev sestavlja večje število naprav (npr. magneti in napajalniki), ki pospešujejo delce do končne točke, ko se delci zaletijo v tarčo. Za nadzorovanje naprav skrbi kontrolni sistem Experimental Physics and Industrial Control System (EPICS).

V fazi razvoja pospeševalnika delcev je potrebno naprave kupiti ali narediti ter jih podpreti v samem kontrolnem sistemu. Ko kontrolni sistem pozna napravo, lahko vso konfiguracijo izvajamo preko EPICS zapisov.

Nastavitve naprav, ki omogočajo optimalno delovanje pospeševalnika, so odvisne od posameznih naprav, celotnega sistema ter preferenc raziskovalcev in jih je analitično zelo težko določiti. To pomeni, da mora imeti osebe, ki upravlja s pospeševalnikom, na voljo enostavne uporabniške vmesnike, ki omogočajo konfiguracijo naprav skozi celotno življenjsko dobo pospeševalnika.

Dosedanja orodja za izdelavo grafičnih uporabniških vmesnikov, ki jih lahko uporabimo za konfiguriranje kontrolnega sistema, so zastarela in zahtevna za uporabo. Poleg tega pa omogočajo nastavljanje le nekaj parametrov naenkrat, kar je za veliko primerov premalo, saj imajo naprave tudi nekaj

sto parametrov, ki jih raziskovalci in razvijalci lahko urejajo. Do sedaj so morali v primerih, ko je bilo potrebno popraviti veliko število parametrov na napravi, posredovati razvijalci oziroma integratorji naprav, ki so se morali fizično priklopiti na Input Output Controller (IOC), na katerega je bila priključena naprava, ter v EPICS zapisih ročno popraviti vrednosti. Primer arhitekture sistema prikazuje Slika 5.1. Če je bilo naprav istega tipa več in so bile priključene na različne IOC-je, so morali postopek ponoviti na vsakem IOC-ju. Delo je bilo zahtevno, časovno potratno ter je ponavadi privedlo do veliko napak.

Problem je opazilo več razvijalskih skupin in posledično se je pojavilo več rešitev, ki so bolj ali manj uspešno reševale omenjen problem.

Cilj diplomskega dela je bil implementirati sklop orodij, ki poenostavlja proces konfiguracije naprav v kontrolnem sistemu. Ob koncu projekta pa se je zastavilo vprašanje, ali bi lahko za konfiguracijo naprav uporabili katerega od obstoječih standardov na tem področju ter kakšne so vzporednice standardnega načina upravljanja omrežij ter implementirane rešitve.

Diplomsko delo tako sestoji iz kratkega opisa pospeševalnika delcev *European Spallation Source*, osnov teorije upravljanja omrežij, opisa implementacije konfiguratorja kontrolnega sistema ter diskusije, kako bi uporaba teorije upravljanja omrežij vplivala na samo implementacijo rešitve.

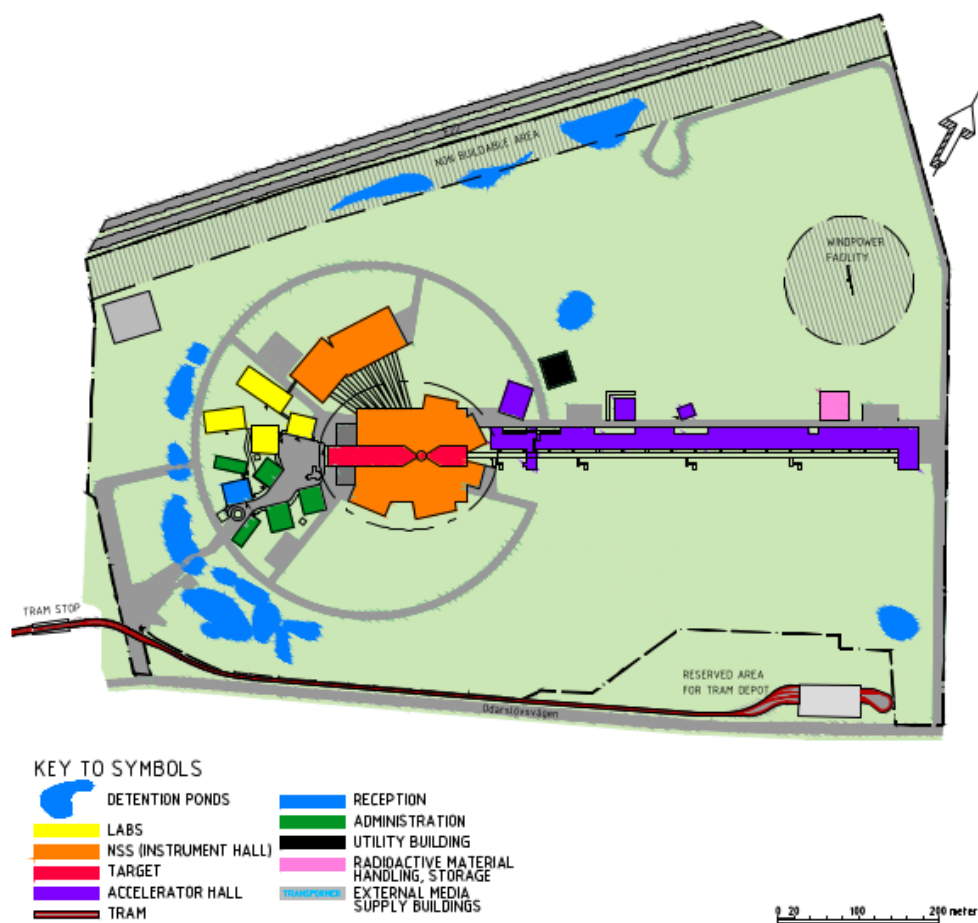
## Poglavje 2

# Upravljanje pospeševalnika European Spallation Source

### 2.1 Pospeševalnik European Spallation Source

European Spallation Source (ESS) je raziskovalna ustanova, v kateri bodo znanstveniki z akademskega področja in industrije raziskovali različna znanstvena področja s pomočjo nevtronskih žarkov. Načrtovan razpored glavnih komponent prikazuje Slika 2.1. ESS bo polno delujoč leta 2025, leta 2013 pa se bo na Švedskem v mestu Lund začela gradnja samega objekta. Metoda sipanja nevtronov omogoča vpogled v molekularno strukturo materialov, ki ni mogoč z uporabo drugih metod. Uporablja se lahko za raziskave na področju fizike, kemije, biologije, raziskave materialov, tehnike in na drugih področjih.

Glavni del ESS bo linearni pospeševalnik ali linac. Njegova naloga bo ustvarjati protone v ionskem izvoru, pospeševati protone, da dosežejo primerno energijo ter jih pripeljati do tarče, kjer se s procesom sipanja ustvarijo nevtroni. Glavni cilj pospeševalnika bo čim učinkovitejše pretvarjanje visoko energijskega žarka protonov iz pospeševalnika v nizko energijske žarke nevtronov [4].



Slika 2.1: Načrtovan razpored glavnih komponent pospeševalnika delcev European Spallation Source

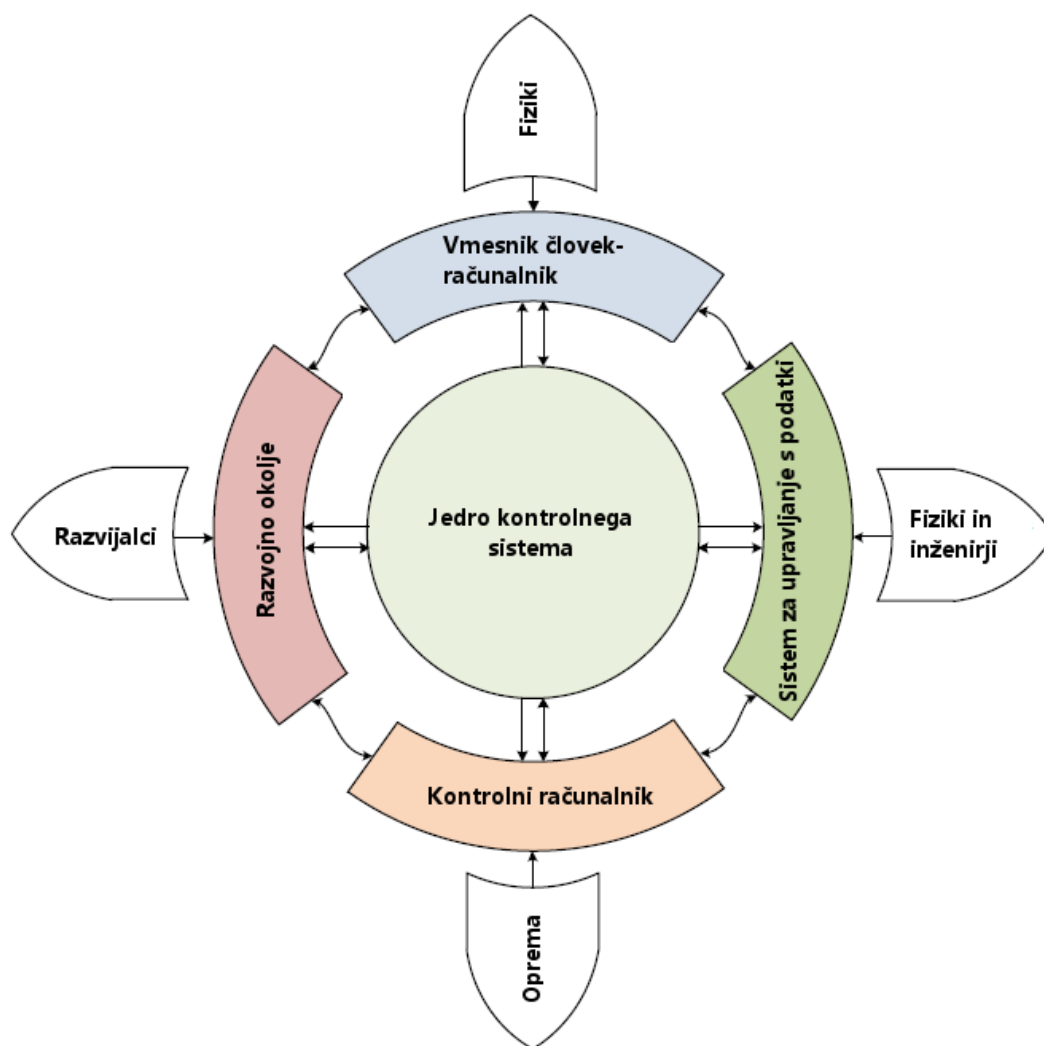
## 2.2 Integrirani kontrolni sistem

Integrirani kontrolni sistem bo odgovoren za celoten ESS. Nadzoroval bo pospeševalnik, tarčo, znanstvena orodja in samo zgradbo. V fazi načrtovanja je ESS izbral obstoječ kontrolni sistem EPICS, katerega razvoj sta začela laboratorija Argonne in Los Alamos. Komponente integriranega kontrolnega sistema so jedro kontrolnega sistema, kontrolni računalniki, sistem za upravljanje s podatki in vmesnik človek-računalnik. Slika 2.2 prikazuje, kako se komponente med seboj povezujejo.

*Jedro kontrolnega sistema.* Jedro kontrolnega sistema je skupek sistemov in orodij, ki inženirjem, tehnikom in fizikom zagotavljajo potrebne podatke, informacije in storitve. Glavne komponente jedra kontrolnega sistema so časovni sistem, ki omogoča sinhronizacijo ure, sistem za varovanje naprav in osebja in nabor storitev kontrolnega sistema.

*Kontrolni računalniki.* Kontrolni računalniki so strežniki, ki nadzirajo nabor opreme. Integrirani kontrolni sistem bo vseboval več nadzornih računalnikov, ki so lahko dodeljeni enemu dobavitelju ali razvijalcu, na primer: interni skupini, sodelujočemu inštitutu ali komercialnemu prodajalcu. Kontrolni računalnik je sestavljen iz standardizirane strojne opreme, komponent, razvojne opreme in storitev. Na zgornji plasti povezuje jedro kontrolnega sistema z vmesnikom človek-računalnik, na spodnji plasti pa z analognimi in digitalnimi signali med seboj povezuje opremo.

*Sistem za upravljanje s podatki.* Centralni sistem za upravljanje s podatki je skupek podatkovnih baz, orodij in storitev, ki hranijo, upravljajo in dostopajo do podatkov. Vsebuje vitalne konfiguracije kontrolnega sistema in podatke fizikalne narave (*lattice*) o pospeševalniku, tarči in znanstvenih orodjih. Sistem za upravljanje s podatki poenostavlja razvoj in pospešuje cikel kodiranja, testiranja in razhroščevanja.



Slika 2.2: Glavne komponente kontrolnega sistema



*Vmesnik človek-računalnik.* Vmesnik je nujen za zagotavljanje dobre uporabniške izkušnje. Obsega širok nabor naprav in programskih orodij. Služiti mora uporabnikom s širokim naborom znanja in izkušenj iz različnih ozadij [4].

## 2.3 Experimental Physics and Industrial Control System (EPICS)

EPICS je okolje za razvoj in implementacijo porazdeljenih kontrolnih sistemov za upravljanje naprav, kot so pospeševalniki delcev, teleskopi in drugi veliki eksperimenti. Orodje je v pomoč pri razvoju sistemov, ki vsebujejo večje število računalnikov v omrežju. Nad računalniki omogoča nadzor in sprejema povratne informacije.

Za komuniciranje med računalniki uporablja dve tehnologiji, in sicer strežnik-odjemalec in objava-naročanje. Prvi sklop računalnikov, ki jim pravimo tudi vhodno-izhodni kontrolerji ali IOC-ji, iz priključenih naprav zbirajo eksperimentalne in kontrolne podatke v realnem času. Te podatke nato pošljejo drugemu sklopu računalnikov (odjemalcem) z uporabo Channel Access (CA) omrežnega protokola. CA protokol je omrežni protokol, ki omogoča visoko pretočnost podatkov in je primeren za aplikacije, ki tečejo v realnem času, kot so aplikacije, ki se uporabljajo pri znanstvenih eksperimentih. EPICS je navadno nameščen na računalnikih, ki so v večini primerov kar navadni računalniki, ki omogočajo povezovanje z različnimi orodji in napravami. Vsak IOC ima nameščene baze EPICS zapisov, ki predstavljajo upravljane naprave. Drugi računalniki v omrežju lahko komunicirajo z IOC-jem preko koncepta kanalov (npr. vhodni kanal, izhodni kanal in analogni vhodni kanal). Z računalniki, na katerih je nameščen kontrolni sistem EPICS, lahko komunicira vsaka naprava, ki implementira protokol CA. Operacije se izvajajo preko ukazne vrstice ali grafičnih uporabniških vmesnikov [5].

### 2.3.1 EPICS zapisi

Obstaja več tipov različnih EPICS zapisov. Primer EPICS zapisa tipa Calc prikazuje Izpis 2.1. Poleg že definiranih tipov zapisov je mogoče ustvariti tudi nove. Vsak EPICS zapis je sestavljen iz več različnih polj, ki vsebujejo različne vrednosti. Polja so namenjena različnim opravilom in so lahko standardna vsem zapisom, določeni zapisi pa imajo definirana tudi svoja lastna polja. Vhodna in izhodna polja omogočajo povezovanje zapisov med seboj. Na ta način se pri izvajanju zapisov vrednosti prenašajo med povezanimi zapisi.

Za primer vzemimo EPICS zapise: Analog Input (AI), Analog Output (AO), Calc in Calcout. AI in AO zapisa shranjujeta oziroma vračata analogni vrednosti in se ponavadi uporabljata za nastavljanje točk, temperatur, pritiska itd. Calc in Calcout zapisa na vhodna polja dobijo vrednosti povezanih zapisov, nad vrednostmi izvedeta matematične operacije in rezultat vrne na izhodno polje.

```
record(aSub, "$(MODULE_INSTANCE_NAME):CALC")
{
    field(DESC, "BPM logic")
    field(INAM, "bmpLogicInit")
    field(SNAM, "bmpLogicProcess")
    field(EFLG, "ALWAYS")

    field(INPA, "$(DAQ_X1) PP")
    field(NOA, "$(NELM)")
    field(FTA, "DOUBLE")
    field(INPB, "$(DAQ_X2) PP")
    field(NOBB, "$(NELM)")
    field(FTB, "DOUBLE")
    field(INPC, "$(DAQ_Y1) PP")
    field(NOC, "$(NELM)")
    field(FTC, "DOUBLE")
    field(INPD, "$(DAQ_Y2) PP")
    field(NOD, "$(NELM)")
}
```

```
    field(FTD, "DOUBLE")
    field(INPE, "$(MODULE_INSTANCE_NAME):SMO0")
    field(FTE, "DOUBLE")

    field(FTVA, "DOUBLE")
    field(NOVA, "$(NELM)")
    field(FTVB, "DOUBLE")
    field(NOVB, "$(NELM)")
    field(OUTC, "$(MODULE_INSTANCE_NAME):X PP")
    field(FTVC, "DOUBLE")
    field(OUTD, "$(MODULE_INSTANCE_NAME):Y PP")
    field(FTVD, "DOUBLE")
}
```

Izpis 2.1: Primer EPICS zapisa tipa Calc

## 2.4 Upravljanje kontrolnega sistema

Sistem za upravljanje s podatki se deli na več delov: varovanje, sistem za upravljanje z različicami, konfigurator kontrolnega sistema, sistem za upravljanje z inventarjem itd. V diplomski nalogi smo se osredotočili na konfigurator kontrolnega sistema.

Konfigurator kontrolnega sistema je sklop orodij in storitev, ki se uporabljajo za shranjevanje podatkov v podatkovne baze in za upravljanje ustreznih sistemov porazdeljenega kontrolnega sistema. Kontrolni računalniki in strežniki so vozlišča računalnikov z unikatnim imenom ter množico programskih paketov, ki morajo biti nameščeni, da lahko računalniki opravljajo svojo vlogo v sistemu. V paketih so operacijski sistem, gonilniki naprav, aplikacije, podatkovna baza ter podpora naprav. Primer implementacije konfiguratorja kontrolnega sistema za pospeševalnik delcev ESS je opisan v Poglavju 4 [4].



## Poglavje 3

# Osnove upravljanja omrežij

### 3.1 Splošno

Upravljanje omrežij lahko definiramo kot izvajanje aktivnosti, administracije, vzdrževanja in zagotavljanja omrežij ter storitev. Aktivnosti predstavljajo dnevne aktivnosti pri zagotavljanju omrežnih storitev, administracija se ukvarja z vzpostavitvijo in nadzorovanjem ciljev, pravil in procedur upravljanja omrežij. Vzdrževanje obravnava nastavitve in vzdrževanje arhitekture in opreme. Zagotavljanje omrežij in storitev vključuje konstrukcijo in načrtovanje omrežij.

Cilj upravljanja omrežij je uporabnikom omrežij zagotoviti kvaliteto informacijskih storitev, kot jo pričakujejo. Za dosego tega cilja mora ponudnik z uporabnikom skleniti dogovor o kvaliteti storitev (SLA). S strani podjetja upravljanje omrežij predstavlja strateško in taktično načrtovanje aktivnosti, administracije, vzdrževanja in zagotavljanja storitev za dosego trenutnih in bodočih ciljev za minimalno ceno.

Poglavje se začne z opisom standardov na področju upravljanja omrežij in modelom upravljanja omrežij, kot ga definira organizacija ISO. Skozi dobro strukturiran model upravljanja omrežij je nato razložen protokol SNMP, kot je bil definiran v svoji prvi različici. Poglavje se zaključi z opisom novosti in izboljšav, ki sta jih prinesli novejši različici protokola SNMP, in sicer SNMPv2

in SNMPv3 [2].

## 3.2 Standardi

V uporabi je več standardov za upravljanje omrežij. To so OSI model, Internet model, TMN, IEEE LAN/MAN in spletno upravljanje.

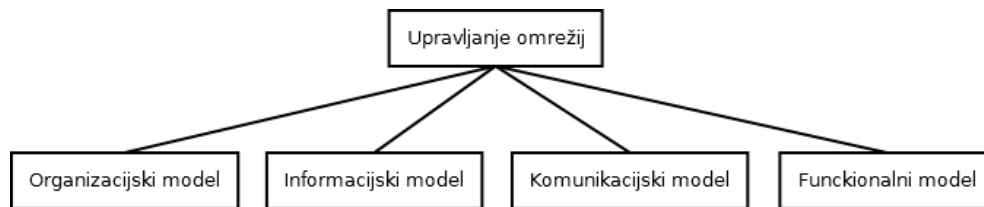
OSI standard za upravljanje omrežij je standard prevzet s strani organizacije ISO. OSI protokol za upravljanje omrežij je CMIP z vgrajenimi storitvami CMIS, ki specificira osnovne storitve za opravljanje najrazličnejših funkcij. Je najbolj obširen spisek specifikacij in obravnava vseh sedem plasti modela OSI. Specifikacije so objektno naravnane in tako opisi naprav temeljijo na razredih in pravilih dedovanja. Dve največji pomanjkljivosti OSI standarda za upravljanje omrežij sta njegova kompleksnost in velikost CMIP sklada.

V nasprotju s CMIP protokolom je SNMP resnično enostaven. Svojo pot je začel kot industrijski standard in skozi leta postal podoben standardnim specifikacijam, kot jih določajo organizacije za postavljanje standardov. Za vse Internet specifikacije je odgovoren IETF, vključno z upravljanjem omrežij. Danes je na največ napravah zaradi svoje enostavnosti implementiran ravno protokol SNMP. V nadaljevanju se bomo zato osredotočili predvsem na protokol SNMP.

TMN je bil načrtovan za upravljanje telekomunikacijskih omrežij in je tako tudi orientiran. TMN je standard organizacije ITU in temelji na OSI CMIP/CMIS specifikacijah. TMN razširja koncept upravljanja omrežij in obravnava tudi poslovni vidik upravljanja omrežij.

IEEE specifikacije za LAN in MAN se ukvarjajo le s prvo (fizično) in z drugo (povezovalno) OSI plastjo in so strukturirane podobno kot OSI specifikacije. OSI/CMIP in Internet/SNMP protokola uporabljata IEEE standarde za spodnje plasti.

Spletno naravnano upravljanje temelji na uporabi spletnih tehnologij. Upravljanje temelji na Internet standardih in na DMTF standardih. Čeprav ime govori o tem, da je upravljanje spletno naravnano pa ni nujno vezano na



Slika 3.1: Model upravljanja omrežij

določen uporabniški vmesnik [1].

### 3.3 Model upravljanja omrežij

OSI model je ISO standard in je izmed vseh omenjenih standardov najbolje specificiran. Je dobro strukturiran in vključuje vse vidike upravljanja omrežij. Slika 3.1 prikazuje OSI model upravljanja omrežij, ki vsebuje: organizacijski model, informacijski model, komunikacijski model in funkcionalni model.

*Organizacijski model* opisuje komponente v upravljanem omrežju, njihove funkcije in razmerja. Definira izraze objekt, agent in upravitelj. Omrežni elementi so lahko omrežni gradniki kot na primer usmerjevalniki, stikala, strežniki. Gradniki so lahko upravljani ali pa ne. Na upravljanih elementih teče upravljalni proces, imenovan agent. Upravitelj komunicira z agentom v upravljani napravi in jo tako upravlja. Upravitelj izvaja vprašanja na agentu. Ko mu jih agent izroči, jih upravitelj obdela in shrani v bazo podatkov. Omrežni element lahko opravlja tudi obe nalogi, nalogo upravnika in agenta. Kot upravitelj zbira podatke omrežnih elementov, jih obdela in rezultate shrani v bazo. Kot agent prenaša informacije upravitelju, ki je nivo višje. Tak tronivojski sistem se lahko uporabi za statistične meritve na omrežju in pošiljanje rezultatov upravitelju na višjem nivoju, ko jih ta potrebuje.

*Informacijski model* se ukvarja s strukturo in z organizacijo informacij. Specificira strukturo informacij (SMI) in bazo informacij (MIB). SMI opisuje, kako so informacije strukturirane, MIB pa, kako se informacije med

seboj povezujejo in kako se shranjujejo. MIB uporabljata upravitelj in agent za shranjevanje in izmenjavo informacij. MIB upravitelja vsebuje vse informacije omrežnih naprav, ki jih upravlja, MIB agenta pa vsebuje le svoje lokalne informacije. Upravitelj ima tako upravljalno bazo podatkov in MIB bazo. Upraviteljeva baza podatkov je prava baza podatkov, ki vsebuje izmerjene ali določene vrednosti naprav v omrežju, MIB navidezna baza pa vsebuje le informacije, ki omogočajo izmenjavo in pridobivanje vrednosti na napravah. Za MIB, ki vsebuje podatke o upravljanem objektu, ni potrebno, da je omejen na fizične elemente, ampak se lahko razširi tudi na podatke o programski opremi ali administrativne podatke (kontaktna oseba, številka računa ...).

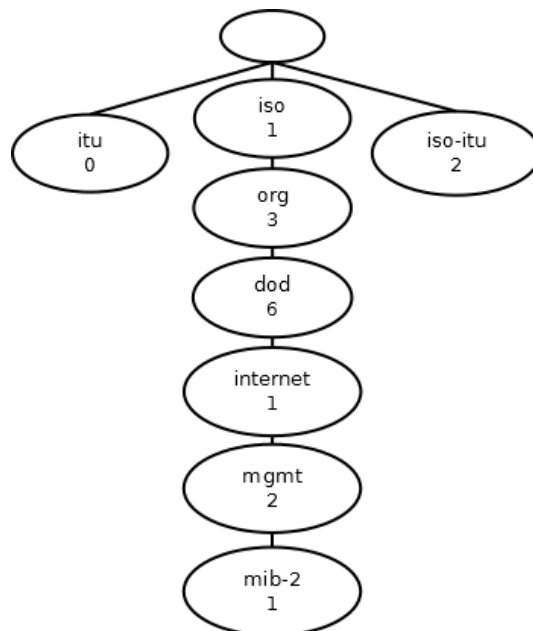
Upravljeni objekti so unikatno definirani v drevesni strukturi (MIT), specificirani v OSI modelu in se uporabljajo v Internet modelu. Vsakemu upravljanemu objektu pripada eno vozlišče v upravljalnem drevesu. Slika 3.2 prikazuje internacionalno uporabljen OSI MIT. Številka v vozlišču identificira objekt na vsakem nivoju posebej, tako je *iso* označen kot 1, *org* kot 1.3, *dod* kot 1.3.6 itd. Alternativna oblika zapisa vozlišč *internet* in *mib-2* se nahaja v Poglavju 3.4.2 [1].

V Internet specifikacijah [6] je vsak upravljeni objekt definiran z naslednjimi parametri:

- *object identifier* in *descriptor* (unikaten ID),
- *syntax* (sintaksa za modeliranje objekta),
- *access* (pravice za dostop do objekta),
- *status* (zastavica, ki pove, ali je implementacija zahtevana ali ne),
- *definition* (opis semantike tipa objekta).

*Komunikacijski model* vključuje prenos informacij med upraviteljem in agentom ter med različnimi upravitelji. Obravnava tudi sestavo sporočil ter povezavo s transportnimi protokoli na nižji plasti.



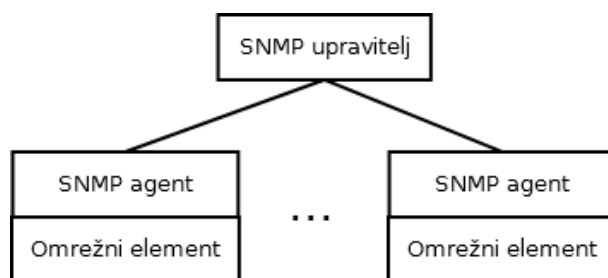


Slika 3.2: OSI MIT

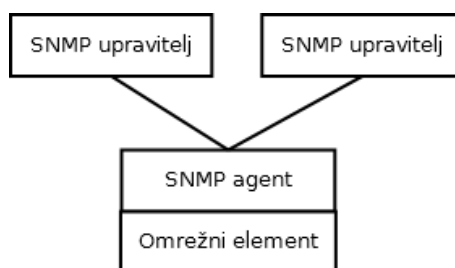
Četrty model se ukvarja z uporabniško orientiranimi zahtevami upravljanja z omrežji. OSI definira naslednje funkcionalnosti: konfiguracije, napake, zmogljivosti, varnost in obračunavanje. IETF eksplicitno ne definira arhitekture za SNMP model upravljanja, obstaja pa implicitno. *Funkcionalni model* naslavlja v smislu operacij, administracije in varnosti.

## 3.4 SNMP

IETF ima nalogo razviti Internetne standarde, med drugimi tudi standard za upravljanje omrežij. Tako je leta 1970 razvil protokol SNMP. SNMP upravljanje omrežij je daleč najbolj razširjeno in največ omrežnih komponent ima vgrajene agente, ki se znajo odzvati na SNMP zahteve. Ko v SNMP upravljano omrežje dodamo novo napravo, ki ima vgrajenega SNMP agenta, jo lahko NMS začne avtomatično nadzorovati. Poznano je tudi kot Internetno upravljanje in vsako omrežje, ki uporablja TCP/IP protokole, je primerno za



Slika 3.3: Dvonivojski model z upraviteljem in večjim številom agentov



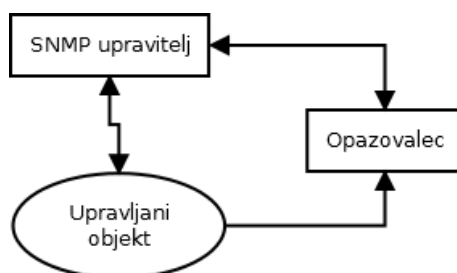
Slika 3.4: Dvonivojski model z dvema upraviteljema

SNMP upravljanje. Preko *proxy* agentov lahko SNMP upravlja tudi omrežja, ki ne temeljijo na TCP/IP tehnologiji.

Enostavnost dodajanja novih naprav v obstoječe upravljano omrežje in konfiguriranje le teh je prispevalo k prepoznavnosti in množični uporabi SNMP-ja. Ker se OSI standard zaradi svoje kompleksnosti ni začel uporabljati, so se SNMP specifikacije dopolnile in nastal je SNMPv2 in SNMPv3. SNMP je bil mišljen kot začasna rešitev, na dolgi rok naj bi se uporabljal standard OSI CMIP/CMIS. SNMPv2 je bil razvit, da bi postal neodvisen od OSI standarda, dodane so mu bile tudi nove funkcionalnosti. SNMPv3 vključuje tudi varnostne popravke, ki so bili v SNMPv2 izpuščeni [1, 2, 3].

### 3.4.1 Organizacijski model

Osnovni organizacijski model je enostaven dvonivojski model. Sestavlja ga proces agenta v upravljanem objektu in proces upravitelja, ki se nahaja



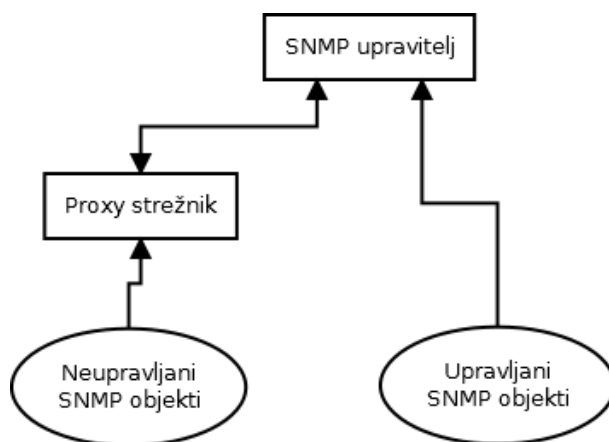
Slika 3.5: Tronivojski model z upraviteljem, agentom in opazovalcem

v omrežnem upravljalnem sistemu in upravlja objekt, kot je prikazano na Sliki 3.3. Upravitelj in agent sta programska modula. Agent se odziva na vsak upravljalni sistem, ki z njim komunicira z uporabo protokola SNMP. Kot je prikazano na Sliki 3.4, lahko več upraviteljev nadzira enega agenta.

V dvonivojskem modelu upravitelj prejema agentove neobdelane podatke in jih obdela. Namesto da upravitelj neprestano prejema in obdeluje podatke, lahko med agenta in upravitelja namestimo vmesnega opazovalca. Opazovalec prejema podatke, jih obdela in obdelane pošlje upravitelju. Taka organizacija privede do tronivojskega modela, ki ga prikazuje Slika 3.5. Upravitelj tako prejema podatke obeh agentov.

Čisti SNMP upravljalni sistem sestoji iz SNMP agentov in SNMP upraviteljev. Poleg upravljanih objektov lahko SNMP upravitelj upravlja tudi objekt, ki nima SNMP agenta. V tem primeru lahko upravitelj upravlja agenta preko *proxy* strežnika. Shemo lahko vidimo na Sliki 3.6. SNMP upravljalni sistem se lahko obnaša kot agent ali upravitelj, kar je podobno arhitekturi strežnik – odjemalec, kjer lahko gostitelj deluje kot strežnik ali odjemalec.

Zaradi enostavnosti protokola SNMP se na transportni plasti največkrat uporablja protokol UDP. Komunikacija med upravljanimi objekti je realizirana z uporabo petih sporočil. Tri sporočila (*get-request*, *get-next-request* in *set-request*) tvori upravitelj, ostali dve (*get-response* in *trap*) pa agent. V SNMP shemi upravitelj nadzoruje omrežje tako, da od agentov zahteva podatke o stanju in karakteristikah, učinkovitost pa se poveča, ko agenti tvorijo



Slika 3.6: Upravljanje agenta preko *proxy* strežnika

alarmna sporočila.

*Get-request* sporočilo tvori upravitelj, ko potrebuje vrednost objekta.

*Get-next-request* sporočilo je zelo podobno sporočilu *get-request* in vrača naslednjo vrednost istega objekta. V več primerih ima lahko objekt več instanc in tako več vrednosti.

*Set-request* tvori upravitelj, ko želi inicializirati ali ponastaviti vrednost nekega objekta.

*Get-response* tvori agent po prejetju sporočil *get-request*, *get-next-request* ali *set-request*. V sporočilo se vstavi vsak uspešen odziv ali pa sporočilo o napaki.

Drugo sporočilo, ki ga tvori agent, je obvestilo (*trap*). Sporočilo se ustvari, ko na primer agent opazi prekoračeno vrednost ali ob drugih pomembnih dogodkih. SNMP upravitelj, ki je del upravljalnega sistema, ima podatkovno bazo, ki vsebuje dve množici podatkov. Prva množica vsebuje informacije o objektih in druga vrednosti teh objektov. MIB je baza navideznih, statičnih podatkov in mora biti prisotna, preden v sistem dodamo nov element. MIB mora biti prisoten tako v upravitelju kot v agentu, da si lahko izmenjata

vse vrednosti objektov. Druga množica podatkov so izmerjene vrednosti objektov. Ta podatkovna baza se lahko implementira v poljubni arhitekturi. Agent za razliko od upravitelja vsebuje le navidezno bazo podatkov MIB [3, 1].

### 3.4.2 Informacijski model

Informacijski model se ukvarja s strukturo informacij o upravljanju (SMI) in z bazo informacij o upravljanju (MIB). Kot osnova za zapis informacij o upravljanju se uporablja sintaksa ASN.1. V MIB-u se nahajajo informacije o združevanju in povezavi objektov v upravljanem sistemu. IETF definira splošne objekte, ki jih lahko upravlja vsak SNMP kompatibilen sistem. Tip objekta je sestavljen iz imena, sintakse in sheme zapisa. Ime je predstavljeno unikatno z deskriptorjem in identifikatorjem objekta, sintaksa tipa objekta je definirana z ASN.1. Kot shema zapisa za prenos podatkov med upraviteljem in agentom se uporablja *Basic Encoding Rules* (BER).

#### SMI

*Imena.* Vsak tip objekta je enolično določen z deskriptorjem (*description*) in identifikatorjem objekta (*object identifier*).

$$\textit{internet OBJECT IDENTIFIER} ::= \{1\ 3\ 6\ 1\}$$

Tudi MIB-II, ki je razširitev MIB-I, se nahaja v Internet MIT.

$$\textit{mib-2 OBJECT IDENTIFIER} ::= \{1\ 3\ 6\ 1\ \textit{mgmt}(2)\ 1\}$$

Zapisa predstavljata alternativno obliko zapisa elementov v MIT. Slika 3.2 prikazuje elementa v drevesni strukturi.

*Sintaksa.* Strukture tipov objektov so definirane z uporabo ASN.1 sintakse, vendar pa v SNMP upravljanju niso definirani vsi konstrukti. SNMP ASN.1 definira le tri kategorije tipov: enostavni tipi, določeni tipi in strukturirani

Struktura	Podatkovni tip
Enostavni tipi	INTEGER
	OCTET STRING
	OBJECT IDENTIFIER
	NULL
Določeni tipi	NetworkAddress
	IpAddress
	Counter
	Gauge
	TimeTicks
	Opaque
Strukturirani tipi	SEQUENCE
	SEQUENCE OF

Tabela 3.1: Tabela prikazuje podatkovne tipe, kot jih definira SNMP ASN.1

tipi. Tipe prikazuje Tabela 3.1. INTEGER se uporablja za zapis števil, OCTET STRING za zapis binarnih ali znakovnih podatkov dolžine 8 bitov, OBJECT IDENTIFIER je namenjen zapisu pozicije objekta znotraj MIB-a in pa NULL, ki določa prazno mesto. NetworkAddress se ne uporablja, IpAddress se uporablja za zapis IP naslova, Counter lahko uporabimo za definiranje vrednosti, ki se vedno povečujejo, Gauge lahko uporabimo za zapis vrednosti, ki se lahko povečujejo ali zmanjšujejo. TimeTicks meri stotinke sekunde oziroma zadnje inicializacije na 0, Opaque pa se uporablja za izgradnjo novih tipov. SEQUENCE in SEQUENCE OF sta strukturirana tipa, ki definirata seznam in tabelo.

*Shema.* SNMP uporablja BER z značko, dolžino in vrednostjo (TLV) za kodiranje informacij, ki se pošiljajo med agentom in upraviteljem. Značke, ki jih uporablja SNMP, so, na primer: BEGIN, CHOICE, END, STATUS itd.

Kot rečeno, ima upravljani objekt pet parametrov: ime, sintaksa, definicija, dostop in status. Specifikacija objekta, ki opisuje sistem, je vidna v Izpisu 3.1. V navedenem primeru je ime predstavljeno kot sysDescr.

```
OBJECT :  
    sysDescr: {system 1}  
    Syntax: OCTET STRING  
    Definition: "Tekstovni opis vnosa"  
    Access: read-only  
    Status: mandatory
```

Izpis 3.1: Specifikacija objekta, ki opisuje sistem

Da lahko zgornjo specifikacijo izvršujemo na računalnikih, mora biti definirana formalno. To naredimo z uporabo maktrojev.

## MIB

MIB je organiziran na način, da lahko implementiramo le tisti del, ki ga potrebujemo. Z uporabo MIB-a dostopamo do upravljanih objektov. Struktura MIB modula je sestavljena iz imena modula, ukazov za uvoz drugih modulov in definicij trenutnega modula. Osnovna ASN.1 struktura je prikazana v Izpisu 3.2 [1].

```
<module name> DEFINITIONS ::= BEGIN  
    <imports>  
    <definitions>
```

Izpis 3.2: Struktura MIB modula

### 3.4.3 Komunikacijski model

Komunikacijski model specificira štiri vidike SNMP komunikacije: arhitekturo, administrativni model za dostop do podatkov, SNMP protokol in SNMP MIB. V nadaljevanju so opisani prvi trije vidiki.

## SNMP arhitektura

SNMP arhitektura specifikira upravljalna sporočila med upravljalnim sistemom in upravljanimi mrežnimi elementi ali objekti. Za prenos informacij med upraviteljem in agentom se uporablja SNMP komunikacijski protokol. SNMP arhitektura ima tri cilje. Prvi cilj je minimizacija števila in kompleksnosti upravljalnih funkcij, ki jih izvaja agent. Drugi cilj je zagotavljanje prožnosti, ki je nujna pri razširitvah in tretji cilj je neodvisnost od arhitekture in mehanizmov gostiteljev in omrežnih prehodov.

SNMP sporočila se izmenjujejo z uporabo protokola UDP. Tako SNMP ostane konsistenten s svojo enostavnostjo, s tem pa se tudi zmanjša omrežni promet.

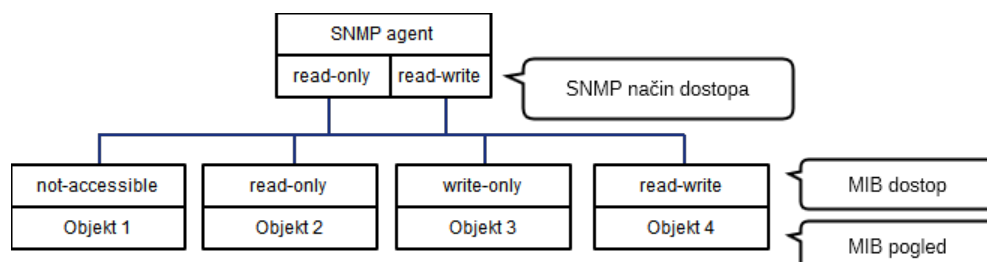
## Administrativni model

Administrativni model se ukvarja z varnostjo ter z dostopom do podatkov.

SNMP upravitelj in SNMP agent skupaj predstavljata *skupnost*. Več parov le-teh lahko pripada isti skupnosti. Različne skupnosti se med seboj ločijo z imenom skupnosti, ki služi tudi za avtentikacijo sporočil, ki si jih člani iste skupnosti pošiljajo med seboj. Omrežni element je sestavljen iz veliko upravljanih objektov in določen agent ima lahko dostop le do nekaterih. To imenujemo *MIB pogled*. Vsaka SNMP skupnost ima določen način dostopa, ki je lahko bralni (*read-only*) ali bralno-pisalni (*read-write*). Tudi MIB pogled ima svoj način dostopa, ki je lahko: nedostopen (*not-accessible*), bralni (*read-only*), pisalni (*write-only*) in bralno-pisalni (*read-write*). Par MIB pogleda in SNMP načina dostopa predstavlja *profil skupnosti*. Profil skupnosti v povezavi z načinom dostopa upravljanega objekta določa operacije, ki jih agenta lahko izvajata na objektu. Kot prikazuje Slika 3.7, lahko SNMP agent z bralno-pisalnim (*read-write*) dostopom izvaja vse operacije nad objekti 2, 3 in 4.

Par SNMP skupnosti in SNMP profila skupnosti predstavlja *SNMP politiko dostopa*, ta pa definira administrativni model SNMP upravljanja. Slika 3.8 prikazuje tri omrežne sisteme, vsak izmed njih ima dostop do različnih do-



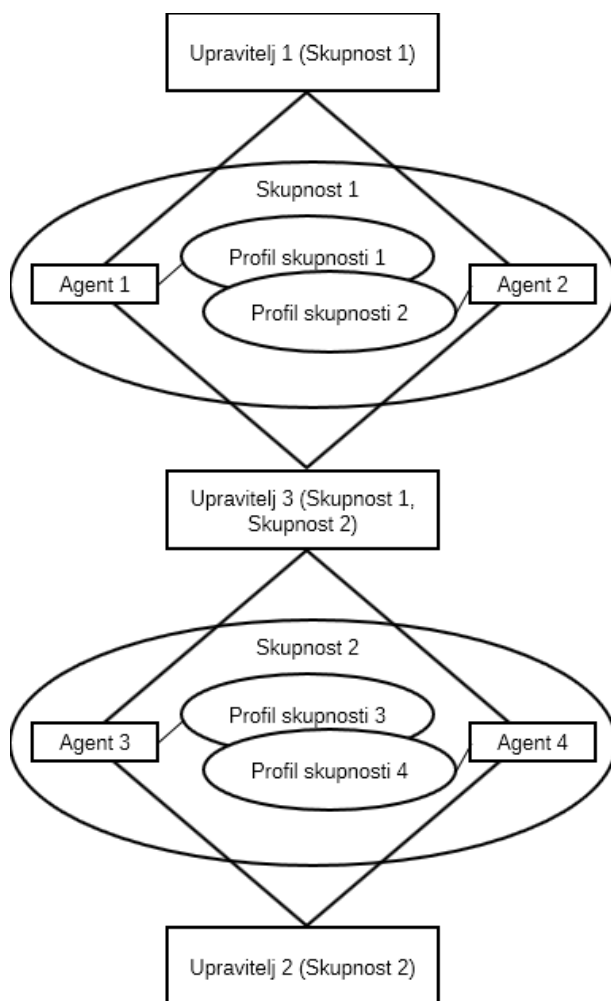


Slika 3.7: Profil SNMP skupnosti

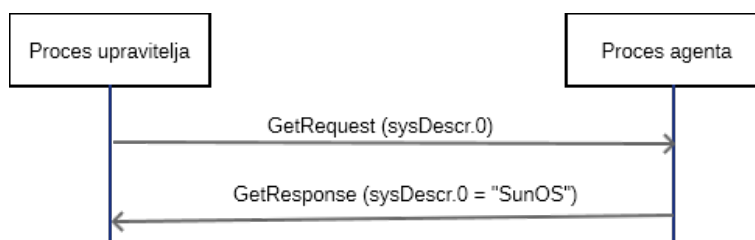
men skupnosti. Agenti 1 in 2 pripadata Skupnosti 1, imata pa različne profile skupnosti. Upravitelj 1, ki je del Skupnosti 1, lahko komunicira z agentoma 1 in 2, ne more pa komunicirati z agentoma 3 in 4, ki pripadata Skupnosti 2. Do agentov 3 in 4 lahko dostopa Upravitelj 2, ne more pa dostopati do agentov 1 in 2. Upravitelj 3 ima dostop do Skupnosti 1 in do Skupnosti 2, kar pomeni, da lahko dostopa do vseh agentov.

### Specifikacija SNMP protokola

Komunikacija med entitetami protokola poteka z uporabo sporočil, vsebovanih v UDP datagramih. SNMP sporočilo je sestavljeno iz različice, imena SNMP skupnosti in podatkovne enote protokola (PDU). Entiteta SNMP protokola se sprejema na vratih 160, Obvestilo (trap) pa na vratih 162. V implementacijah morajo biti nujno podprti vsi tipi: *GetRequest-PDU*, *GetNextRequest-PDU*, *GetResponse-PDU*, *SetRequest-PDU* in *Trap-PDU*. Komunikacija poteka na naslednji način. Entiteta protokola, ki tvori sporočilo, sestavi primeren PDU kot ASN.1 objekt. Nato objekt, skupaj z imenom skupnosti in naslovom vira in ponora, pošlje avtentikacijski shemi. Avtentikacijska shema vrne nov ASN.1 objekt, ki se skupaj z verzijo in imenom skupnosti pošlje naslovniku. Pred tem se sporočilo prerazporedi z uporabo BER.



Slika 3.8: SNMP politika dostopa



Slika 3.9: GetResponse operacija

### SNMP operacije

SNMP operacije so *get* in *set* sporočila, ki se pošiljajo od upravitelja k agentu in *get* in *trap* sporočila, ki se pošiljajo od agenta k upravitelju.

*GetRequest-PDU operacija.* Slika 3.9 prikazuje operaciji, ki omogočata pridobivanje vrednosti objektov. Proces se začne z *get-request* operacijo, ki uporablja *GetRequest-PDU* in ki potuje od upravitelja k agentu. Nato se na agentu izvrši *get-response* operacija, ki uporablja *GetResponse-PDU*.

Ostale operacije se izvajajo podobno kot *GetRequest-PDU* operacija. Posebnost je operacija *GetNextRequest-PDU*, ki se začne z operacijo *GetRequest*, odgovor upravitelj dobi preko operacije *GetResponse*, nato pa upravitelj tvori operacijo *GetNextResponse*, ki vrne naslednji objekt v MIB-u. Objekti so v MIB-u razporejeni leksikografsko po identifikatorju objekta. Operacija je primerna predvsem za izpis vrednosti iz tabel, katerih velikost se spreminja dinamično [3, 2, 1].

#### 3.4.4 Funkcionalni model

SNMPv1 nima formalnih specifikacij funkcij. OSI standard naslavlja pet funkcijskih področij: konfiguracijo, okvare, zmogljivosti, varnost in zaračunavanje storitev. Zaradi omejitve pisanja v upravljane objekte je konfiguriranje objektov omejeno na specifične sisteme ali pa na uporabo terminala, s katerim lahko nastavljamo parametre neposredno na objektu. Upravljanje okvar je implementirano s števcem napak, vgrajenimi v agente. Te lahko upravitelj pre-

bere in obdela. Obvestila (*trap*) se uporabljajo za nadziranje stanja omrežnih elementov in vmesnikov. Števci zmogljivosti so del agentovega MIB-a, s čimer si lahko upravitelj pomaga pri analizi zmogljivosti. Za tako statistično operacijo lahko uporabimo tudi vmesnega upravitelja/agenta, ki implementira protokol RMON. Varnostne operacije so vključene v administrativni model, zaračunavanje pa v protokolu SNMP ni implementirano [3, 2, 1].

### 3.5 SNMPv3

SNMPv1, originalno imenovan SNMP, je potreboval izboljšave in tako je bil razvit SNMPv2. Glavna izboljšava bi morala biti zagotavljanje varnostih funkcij, ki pa so bile tik ob izdaji SNMPv2 izpuščene iz specifikacij. Varnost je bila izboljšana šele v SNMPv3.

SNMPv3 je spremenil protokol le z dodajanjem kriptografskih mehanizmov, vseeno pa izgleda drugače zaradi novih konvencij, konceptov in terminologije. V SNMPv1 in SNMPv2 avtentikacijo predstavlja članstvo v skupnosti, tako da si upravitelj in agent pošiljata ime skupnosti kot navadno sporočilo. Vsako sporočilo SNMPv3 vsebuje varnostne parametre, ki so kodirani kot niz osmih znakov. SNMPv3 zagotavlja zaupnost, integriteto in avtentikacijo pri prenosu sporočil [2].

# Poglavje 4

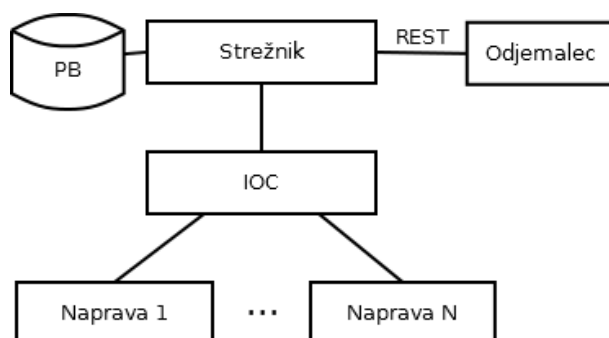
## Izvedba

Sklop programov, imenovan tudi konfigurator kontrolnega sistema, omogoča uvoz konfiguracij novega tipa naprav oziroma tipa naprav, ki ga dodajamo v omrežje kontrolnega sistema, instanciranje novih naprav uvoženega tipa ter spreminjanje konfiguracij na obstoječih instancah. CLI program omogoča uvoz novega tipa naprav v podatkovno bazo, spletna storitev shranjuje in vrača podatke iz podatkovne baze, odjemalec pa podatke prikazuje ter na uporabniku prijazen način omogoča spreminjanje konfiguracij. Arhitekturo sistema prikazuje Slika 4.1.

### 4.1 Razvoj nove naprave

Ko integratorji oziroma razvijalci razvijejo novo napravo, ki se v pospeševalniku delcev še ne nahaja, morajo najprej zagotoviti, da zanjo obstajajo gonilniki ter podpora v kontrolnem sistemu EPICS. V produkcijski postavitvi se bo enaka naprava uporabljala na več mestih. Naprave določenih tipov se lahko v pospeševalniku delcev pojavijo tudi na več sto mestih.

Za novo napravo se pripravi projekt, ki omogoča enostaven razvoj ostalih naprav istega tipa. Te naprave bodo imele zelo podobne nastavitve, razlikovale pa se bodo v določenih lastnostih, ki jih lahko izpostavimo, ostale lastnosti pa se lahko ob ustvarjanju nove instance naprave le skopirajo.



Slika 4.1: Arhitektura sistema. Podatkovna baza (PB) je povezana direktno na strežnik, odjemalec se na strežnik povezuje preko REST spletnega protokola. Na zahtevo uporabnika se konfiguracijske datoteke prenesejo preko strežnika na IOC-je, ki nato upravljajo naprave.

Za lažjo predstavo smo v sklopu implementacije konfiguratorja kontrolnega sistema ustvarili nov projekt in ga poimenovali *TestniTip*, na katerem bo temeljila razlaga same implementacije. V projektu se nahajajo tri pomembne datoteke *init-pre.cmd*, *st.cmd* in *pom.xml*. *Init-pre.cmd* ter *pom.xml* vsebujeta podatke, potrebne za ustvarjanje novega tipa naprave, datoteka *st.cmd* pa je izhodna konfiguracijska datoteka, ki jo neposredno uporablja kontrolni sistem.

#### 4.1.1 *init-pre.cmd*

V *init-pre.cmd* se nahajajo definicije lastnosti tipa naprave. Pravila za zapis definicij smo določili sami. V datoteki lahko definiramo vse lastnosti tipa naprave, ki jih želimo upravljati brez direktnega spreminjanja EPICS zapisov. Da raziskovalcu olajšamo delo, lahko poleg imena in opisa lastnosti definiramo tudi pričakovan tip vrednosti ter dodatne omejitve, kot na primer minimalno in maksimalno število znakov. Izpis 4.1 prikazuje del vsebine datoteke *init-pre.cmd*, ki je del projekta *TestniTip*. V izpisu so definirane tri lastnosti *TestnegaTipa*: *DAQ\_Y2*, *SMOO* in *NELM*.

```
# @field DAQ_Y2
# @type LINK
# @restrictions DataAcquisition
# Signal from data acquisition module (bottom).
#
# @field SM00
# @type DOUBLE
# @restrictions 0..1
# Smoothing filter.
#
# @field NELM
# @type INTEGER
# @restrictions 1..10000
# Size of the waveform produced by the DAQ, which
# is also the size generated by the BPM.
```

Izpis 4.1: Primer vsebine datoteke init-pre.cmd

### 4.1.2 pom.xml

Del projekta razvoja in upravljanja tipa naprave je tudi pom.xml datoteka, v kateri se nahajajo ime tipa naprave, različica tipa naprave ter še nekateri drugi podatki. Izpis 4.2 prikazuje del vsebine pom.xml datoteke.

```
<project>
  <artifactId>TestniTip</artifactId>
  <version>0.0a1</version>

  <repositories />
  ...
</project>
```

Izpis 4.2: Primer vsebine datoteke pom.xml

### 4.1.3 Uvoz tipa naprave v podatkovno bazo

S CLI programom lahko uvozimo nov tip naprave v podatkovno bazo tako, da mu podamo poti do datotek `init-pre.cmd` in `pom.xml`. Program bo nato v podatkovno bazo zapisal nov tip z imenom in različico, navedeno v `pom.xml`. Nato bo razčlenil `init-pre.cmd`, lastnosti povezal s tipom naprave ter jih prav tako shranil v podatkovno bazo. Primer klica CLI programa iz *Linux* ukazne vrstice se nahaja v Izpisu 4.3.

```
bled --import-component-definition pom.xml --pre init-pre.cmd
```

Izpis 4.3: Primer klica CLI programa, imenovanega `bled`, s katerim lahko v bazo uvozimo nov tip naprave

## 4.2 Odjemalec konfiguratorja kontrolnega sistema

Odjemalca konfiguratorja kontrolnega sistema smo implementirali v Microsoft Excelu. Omogoča nam pregled naprav določenega tipa ter dodajanje, spreminjanje in odstranjevanje naprav. Po opravljenih spremembah nam omogoča tvorjenje konfiguracijskih datotek ter ponovni zagon IOC-jev, na katere so bile predhodno prenesene konfiguracijske datoteke. Primer zaslonske slike prikazuje Slika 4.2.

### 4.2.1 Komunikacija odjemalca s strežnikom in podatkovno bazo

Odjemalec komunicira s strežnikom preko REST spletnega protokola, strežnik pa nato neposredno komunicira s podatkovno bazo. Strežnik omogoča naslednje storitve:

- pridobitev seznama vseh tipov naprav,



## 4.2. ODJEMALEC KONFIGURATORJA KONTROLNEGA SISTEMA 31

	B	C	D	E	F	G	H
1							
2							
3	Delete	Name	loc	TYPE	CURRENT	VOLTAGE	PROTECTION
4		PowerSupply_A	ioc1	Programmable	11.0000	12	SCP
5		PowerSupply_B	ioc1	Uninterruptible	12	13	OPP
6		PowerSupply_C	ioc1	High-voltage	13.0000	10	OCP
7		PowerSupply_D	ioc1	Switched-mode	80	9	OTP
8		PowerSupply_E	ioc1	Battery	15.0000	8	OVP
9		PowerSupply_F	ioc1	Programmable	20	7	UVP
10		PowerSupply_G	ioc1	Uninterruptible	30	24	UPS
11	Delete	PowerSupply_E	ioc1	Battery	15	8	OVP
12							

Slika 4.2: Odjemalec konfiguratorja kontrolnega sistema, implementiran v Microsoft Excelu

- pridobitev naprav določenega tipa in
- shranjevanje naprav določenega tipa.

Če želimo pridobiti seznam naprav določenega tipa, iz seznama izberemo želen tip naprave. Naprave se nato prikažejo v tabeli, ki jo prikazuje Slika 4.2. Vsaka vrstica predstavlja instanco naprave, vsak stolpec pa lastnost naprave, ki jo je moč spreminjati. Z dodajanjem nove vrstice v tabelo ustvarimo novo instanco naprave. Ko smo zadovoljni z novimi nastavitvami, lahko spremenjene podatke shranimo nazaj v podatkovno bazo.

### 4.2.2 Komunikacija odjemalca z IOC-ji

Poleg dela s podatkovno bazo nam strežnik omogoča tudi komunikacijo odjemalca z IOC-ji:

- tvorjenje st.cmd datotek za izbrane IOC-je in
- ponovni zagon IOC-jev.

Vsaka naprava ima nekaj lastnosti, ki povedo, kje se naprava nahaja in katerega tipa je. Na primer lastnost IOC predstavlja ime IOC-ja, na katerega je naprava priključena. Ko uporabnik ureja lastnosti naprave, vrednost

Delete	Name	loc	...
	PowerSupply_A	loc1	...
	PowerSupply_B	loc1	...
		loc2	
		loc3	
		loc4	

Slika 4.3: Po kliku na celico se prikaže spustni seznam, ki vsebuje imena vseh trenutno priključenih IOC-jev

lastnosti IOC izbere iz spustnega seznama vseh IOC-jev. Primer prikazuje Slika 4.3.

Ko se uporabnik odloči za izvoz konfiguracij, se vse spremembe najprej shranijo v podatkovno bazo, nato pa strežnik tvori po eno datoteko `st.cmd` za vsak IOC, v katero nato za vsako napravo, ki je priključena na IOC, izpiše lastnosti in nastavljene vrednosti po naslednjem pravilu.

$$epicsEnvSet("lastnost", "vrednost")$$

Na IOC-jih je nameščen operacijski sistem *Linux* in omenjen ukaz za vsako lastnost ustvari okolijsko spremenljivko. Vrednost lastnosti pa postane vrednost okolijske spremenljivke.

Izpis 4.4 prikazuje konfiguraciji za dve napravi tipa *TestniTip*. Obe napravi sta priklopljeni na isti IOC, zato se konfiguraciji nahajata v isti datoteki. Ker sta napravi enakega tipa, imata tudi enake lastnosti, vrednosti lastnosti pa se razlikujejo. Za vrsticami z lastnostmi vsake naprave se nahaja zaganjalnik skripte, ki prebere EPICS zapise za tip naprave, vse okolijske spremenljivke zamenja z vrednostmi ter tako ustvari EPICS zapise za specifično napravo. Ko kontrolni sistem komunicira z napravo, za to uporablja predpripravljene EPICS zapise.

```
...
epicsEnvSet("DAQ_Y2", "DAQ12")
epicsEnvSet("SM00", "0.5")
epicsEnvSet("NELM", "2")
```

```
epicsEnvSet("MODULE_INSTANCE_NAME", "BPM3")
< /.../TestniTip/init-pre.cmd

epicsEnvSet("DAQ_Y2", "DAQ13")
epicsEnvSet("SM00", "0.7")
epicsEnvSet("NELM", "5")
epicsEnvSet("MODULE_INSTANCE_NAME", "BPM4")
< /.../TestniTip/init-pre.cmd
...
```

Izpis 4.4: Primer vsebine datoteke st.cmd

Ko so st.cmd datoteke nameščene, lahko ponovno zaženemo IOC-je. Vsak IOC najprej zažene st.cmd datoteko, ki pripravi EPICS zapise za vse naprave, ki so nanj priključene. Pripravljene EPICS zapise nato uporabi kontrolni sistem za izvajanje svojih nalog.



## Poglavje 5

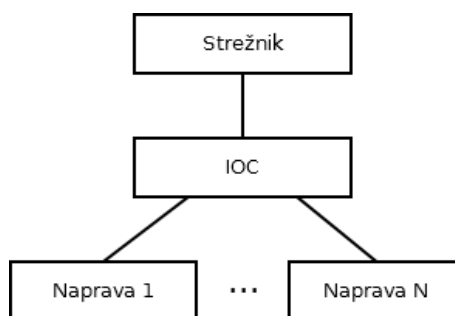
# Rezultati in razprava

Ker smo implementacijo konfiguratorja kontrolnega sistema izvedli z omejenimi sredstvi in v omejenem času, se teoriji in standardom, ki že obstajajo na področju upravljanja naprav, nismo posebej posvetili. Po pregledu teorije pa smo prišli do ugotovitev, ki so navedene v nadaljevanju. Zaradi preglednosti smo se razprave lotili skozi OSI modele upravljanja omrežij. Ker v SNMP upravljanju omrežij zadnji, funkcionalni model ni definiran posebej, ampak je delno definiran v ostalih modelih, smo ga pri razpravi izpustili.

### 5.1 Organizacijski model

V skladu z žargonom upravljanja omrežij smo tudi v našem sistemu, v katerem deluje naša implementacija, določili upravitelja in agente. V našem primeru je strežnik tvoril konfiguracijske datoteke in jih pošiljal na IOC-je, skladno z OSI modelom bi ga poimenovali **upravitelj**. Na drugi strani bi lahko IOC-je, na katerih so priključene naprave, poimenovali **agenti**. Da bomo lažje ločili elemente implementacije od same teorije upravljanja omrežij, bomo še naprej uporabljali izraze strežnik in IOC.

Elemente v našem sistemu smo organizirali v dvonivojski model. Na prvem nivoju je strežnik, na drugem pa več IOC-jev. Postavitev prikazuje Slika 5.1. Strežnik upravlja IOC-je na način, da jim na zahtevo uporab-



Slika 5.1: Dvonivojski model implementiranega sistema

nika pošlje nova sporočila. V trenutni implementaciji strežnik od IOC-jev ne more zahtevati podatkov. Lahko pa tvori sporočila, ki nastavljajo vrednosti na IOC-jih. Prav tako IOC ničesar ne sporoča strežniku, saj se konfigurator kontrolnega sistema ne ukvarja s poizvedovanjem o meritvah ter s shranjevanjem izmerjenih podatkov, temveč le s konfiguracijo naprav, ki omogočajo delovanje sistema. V primeru, da pride do napake pri delovanju pospeševalnika delcev, je potrebno podatke o tem pridobiti iz drugih orodij kontrolnega sistema. Podobno kot SNMP upravitelj ima tudi strežnik podatkovno bazo, v kateri so shranjene vrednosti ter navidezno bazo podatkov, ki pove, katere lastnosti naprav lahko nastavljamo in kakšnega tipa so. Navidezne baze niso specifične za IOC, ampak za naprave, ki so priključene na IOC. V nasprotju s SNMP agentom IOC nima navidezne baze podatkov.

V pogledu na implementacijo skozi organizacijski model bi z implementacijo SNMP protokola pridobili veliko. Strežnik bi lahko z uporabo *get-request* in *get-next-request* operacij pridobival trenutne podatke iz IOC-jev. IOC pa bi lahko v primeru konfiguracijske napake inicializiral operacijo *trap* in tako strežniku omogočil takojšni odziv. Prav tako bi dostop do agentov omogočili vsem upraviteljem, ki implementirajo SNMP protokol.

## 5.2 Informacijski model

V nasprotju s SNMP modelom upravljani objekti niso razporejeni v globalno drevo objektov, ki bi bili lahko dostopni od koder koli in na isti način. Samo strukturo podatkov smo dorekli pred začetkom same implementacije in je specifična tej rešitvi. Imamo navidezno bazo, ki je značilna za vsak tip naprave, vsaka instanca naprave pa je določena z imenom ter ICO-jem, na katerega je priključena. Ko strežnik tvori konfiguracije iz podatkov, ki so shranjeni v bazi, te podatke zapiše v datoteko, nato pa celotno datoteko pošlje IOC-ju. Rešitev ne omogoča nastavljanja vrednosti specifičnih naprav direktno na IOC-ju, s tem pa se poveča tudi promet med strežnikom in IOC-jem.

Z implementacijo SNMP-ja bi bilo potrebno ob izdelavi nove naprave narediti MIB shemo in to bazo uvoziti na strežnik. To bi bil tudi časovno najbolj potraten del. Ko pa bi IOC in strežnik poznala MIB shemo, bi vsa komunikacija potekala na standarden način. Strežnik bi lahko nastavljal specifične vrednosti upravljanih objektov direktno na IOC-jih. S tem bi se na dolgi rok upravljanje poenostavilo, zmanjšal pa bi se tudi omrežni promet.

## 5.3 Komunikacijski model

Problem, ki ga v implementaciji nismo naslovili, je bil problem administracije. Vsak uporabnik, ki ima dostop do strežnika, lahko spreminja vrednosti lastnosti vseh naprav na vseh IOC-jih. Samo avtentikacijo uporabnika bi bilo enostavno implementirati, precej težje pa bi bilo določiti dostope uporabnika do posameznih lastnosti naprav (do katerih naprav ima uporabnik dostop, kakšen je način dostopa do naprav in kakšen je način dostopa do posameznih lastnosti naprav).

SNMP je že v svoji prvi različici ponudil dober model dostopa do posameznih upravljanih objektov, ki je bil v naslednjih dveh različicah še izpopolnjen. Z implementacijo SNMP-ja bi omogočili različne poglede na sistem ter omogočili urejanje različnih skupin lastnosti naprav, ki bi bile primerne za različne kadre, ki upravljajo sam pospeševalnik delcev.





## Poglavje 6

### Zaključek

Z implementacijo smo pokazali, da obstajajo alternativni pristopi h konfiguraciji kontrolnih sistemov ter da je pristop primeren za razvijalce in integratorje, ki morajo imeti možnost nastavljanja več lastnosti naprave hkrati kot tudi za raziskovalce, ki jim uporabniški vmesnik omogoča hitro spreminjanje specifičnih lastnosti naprave ter ponoven zagon ICO-jev, na katere so naprave priključene.

Po zaključku projekta smo se začeli ukvarjati z vprašanjem, ali bi lahko z uporabo obstoječih tehnologij na področju upravljanja omrežij implementacijo poenostavili in jo naredili bolj razširljivo. Ugotovili smo, da bi implementacija SNMP protokola omogočila bolj strukturiran pristop k upravljanju naprav, priključenih na IOC. Strežnik bi lahko proizvedoval o trenutnih vrednostih lastnosti naprav, priključenih na IOC, IOC bi lahko obvestil strežnik, da je prišlo do napake ali do drugih pomembnih dogodkov. Poleg tega bi omogočili upravljanje naprav vsem upraviteljem, ki implementirajo SNMP.

Ker se bo pospeševalnik delcev *European Spallation Source* uporabljal v znanstvene namene, bi standarden pristop k upravljanju naprav na samem pospeševalniku raziskovalcem precej olajšal delo. SNMP implementacija v konfiguratorju kontrolnega sistema bi raziskovalcem omogočila vpogled v pospeševalnik v realnem času in z uporabo standardnih orodij.



# Literatura

- [1] M. Subramanian. *Network Management, Principles and Practise*. Addison-Wesley, 2000.
- [2] L. Walsh. *SNMP MIB Handbook, Essential Guide to MIB Development, Use, and Diagnosis*. Wyndham Press, 2008.
- [3] W. Stallings. *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Addison-Wesley, 1999.
- [4] European Spallation Source. ESS Technical Design Report. [http://eval.esss.lu.se/DocDB/0002/000274/015/TDR\\_online\\_ver\\_all.pdf](http://eval.esss.lu.se/DocDB/0002/000274/015/TDR_online_ver_all.pdf). Pogledano 21. 10. 2013.
- [5] Argonne National Laboratory. Experimental Physics and Industrial Control System. <http://www.aps.anl.gov/epics/>. Pogledano 20. 10. 2013.
- [6] Network Working Group. Structure and Identification of Management Information for TCP/IP-based Internets. <http://www.rfc-editor.org/rfc/rfc1155.txt>. Pogledano 24. 10. 2013.